

SISTEMA DE VENDAS COM CONTROLE DE ESTOQUE

Jeferson Tomas dos Santos
Graduando em Sistemas de Informação – Uni-FACEF
jeferson.tomas.santos@gmail.com

Orientador: Prof. Me. Leandro Borges
Mestre em Desenvolvimento Regional – Uni-FACEF
leandro.borges@facef.br

Resumo

Toda empresa que visa um crescimento exponencial deve investir em um sistema de gerenciamento, pois a boa gestão é de suma importância para o varejista no momento de controlar os processos do negócio, mas nem sempre é isso que acontece por problemas financeiros ou mesmo falta de conhecimento. O objetivo deste artigo é apresentar as fases do desenvolvimento de um sistema de gerenciamento de estoque e vendas onde o usuário conta com cadastro de clientes, produtos e fornecedores e relatórios de vendas, orçamentos entre outros. Para o desenvolvimento do sistema foram usadas tecnologias que garantem uma alta performance e resultados satisfatórios além de contar com toda a documentação de software. Após todo o processo de estudo e desenvolvimento da aplicação, conclui-se que todo o projeto foi finalizado e está funcional, podendo contar com novas implementações futuras.

Palavras-chave: Gestão de negócios. Automação comercial. Vendas.

Abstract

Every company that seeks exponential growth should invest in a management system, because good management is of paramount importance for the moment of controlling business processes, but this is not always what happens due to financial problems or even lack of knowledge. The objective of this article is to present the phases of the development of an inventory and sales management system where the user has a register of customers, products and suppliers and sales reports, budgets, among others. guarantee high performance and satisfactory results in addition to having all software documents. After the entire process of studying and developing the application, it is concluded that the entire project has been completed and is functional and may have new future implementations.

Keywords: Business management. Commercial automation. Sales

1. INTRODUÇÃO

1.1. Definição do Problema

Manter um pequeno negócio funcionando e gerando lucro pode ser difícil e muitas pessoas que querem abrir um pequeno empreendimento encontram muitas dificuldades podendo elas ser financeiras ou falha para gerenciar que pode ser ocasionada por falta de ferramentas adequadas ou até mesmo por falta de conhecimento.

Pequenos empreendedores podem não ter o capital necessário ou apenas não veem a necessidade de ter um sistema muito complexo como o que seria usado em uma empresa de grande porte.

Pensando nisso, surgiu a ideia de desenvolver um sistema de baixo custo, pois uma simples pesquisa de mercado nos mostra valores que vão de R\$49,90 até R\$600,00 para aplicações que tem a mesma finalidade. O sistema desenvolvido conta com a possibilidade de obter um controle de estoque, vendas, clientes e fornecedores além de relatórios.

1.2. Justificativa e Objetivos

Hoje em dia é vital para o bom desenvolvimento de uma empresa de pequeno, médio ou grande porte um sistema para gerenciar seu funcionamento, pois com a má gestão começam a aparecer os problemas que podem se agravar ao longo do tempo. Muitas dessas pequenas empresas têm seus dados diversos gravados em uma planilha eletrônica, um caderno ou até mesmo não existe controle nenhum, e isso dificulta seu bom gerenciamento.

O sistema desenvolvido neste trabalho é uma versão reduzida de um ERP (*Enterprise Resource Planning*) visto que um sistema ERP ajuda o gestor a administrar as contas a pagar e a receber, monitorar as vendas e acompanhar os pedidos de compras, dito isto, a comparação se deve pois o sistema de vendas com controle de estoque integra todos os dados de vendas, clientes, produtos (estoque), fornecedor, e disponibiliza relatórios como, todas as vendas, todas as vendas de um cliente, produtos por fornecedor, dados do cliente.

Planejamento de recursos empresariais (ERP) refere-se a um conjunto de softwares que as organizações usam para gerenciar atividades de negócios diárias, como contabilidade, compras, gerenciamento de projeto, gerenciamento de risco e conformidade, e operações da cadeia de suprimentos. Um pacote completo de ERP também inclui o enterprise performance management, software que ajuda a planejar, orçar, prever e relatar os resultados financeiros de uma organização. (O Que é um ERP ? Oracle Brasil, 2020)

Como já foi dito, o sistema é uma alternativa de baixo custo que visa auxiliar o pequeno empreendedor a gerenciar seu negócio de uma forma mais dinâmica, ajudando-o na tomada de decisões.

1.3. Procedimentos Metodológicos

No desenvolvimento do projeto, foram definidas tecnologias que garantem uma performance, bons resultados e implementação relativamente simples. Para o desenvolvimento da Aplicação Desktop foi escolhido o SDK (*Software Development Kit*) da linguagem de programação Java. O XAMPP como servidor e o Sistema Gerenciador de Banco de Dados (SGBD) MySQL que está integrado ao XAMPP. Para a leitura dos códigos de barra, o APP AndroCode Scanner foi escolhido.

A documentação de software também foi desenvolvida, contemplando requisitos funcionais e requisitos não funcionais, diagrama de caso de uso, diagrama de classe, modelos de 5W1H, TAP e EAP.

1.4. Estrutura do artigo

O artigo constitui-se nas seções a seguir: Introdução; na seção dois o referencial teórico, apresentando os conceitos vitais para o desenvolvimento do projeto; o tema Empreendedorismo será abordado na seção três, apontando como o empreendedor pode tirar vantagem dos dados de vendas a clientes. Na seção quatro é apresentado a análise do projeto onde será mostrado a modelagem do banco de dados e conceitos de engenharia de software aplicados na solução. A análise geral do que foi apresentado até agora, será feito na seção cinco, assim como a apresentação do resultado.

2. REFERENCIAL TEÓRICO

2.1. Tecnologias envolvidas no projeto

2.1.1. Java e NetBeans

Para o desenvolvimento da Aplicação Desktop foi escolhido o SDK (*Software Development Kit*) do Java (na sua versão 8, Update 241), que é a linguagem de

programação mais solicitada, por isso o número de ferramentas que facilita no desenvolvimento é grande, além disso o grupo de usuários é muitíssimo grande sendo muito fácil obter material para estudo.

Um objetivo-chave do Java é ser capaz de escrever programas e serem executados em uma grande variedade de sistemas computacionais e dispositivos controlados por computador. Isso às vezes é chamado de "escreva uma vez, execute em qualquer lugar". (DEITEL & DEITEL, Java Como Programar, 2016, p. 13)

Java implementa características como herança, encapsulamento e várias outras habilidades particulares que a orientação a objetos proporciona, como também as características já conhecidas no mundo da programação, como por exemplo instruções de controle e operadores lógicos (DEITEL, 2010).

[...] a Sun Microsystems, em 1991, financiou um projeto de pesquisa corporativa interna que resultou em uma linguagem baseada em C++ que seu criador, James Gosling, chamou de Oak em homenagem a uma árvore de carvalho vista de sua janela na Sun. Descobriu-se mais tarde que já havia uma linguagem de computador com esse nome. Quando sua equipe visitou uma cafeteria local, o nome Java (cidade de origem de um tipo de café importado) foi sugerido; e o nome pegou. (DEITEL & DEITEL, Java Como Programar, 2010, p. 6)

A IDE escolhida foi o NetBeans, uma IDE (ambiente de desenvolvimento integrado) gratuito e de código aberto para desenvolvedores de software podendo ser executado em muitas plataformas, como Windows, Linux, Solaris e MacOS. O NetBeans IDE oferece aos desenvolvedores todas as ferramentas necessárias para criar aplicativos profissionais de desktop, empresariais, Web e móveis multiplataformas.

O NetBeans foi iniciado em 1996 por dois estudantes tchecos na Universidade de Charles, em Praga, quando a linguagem de programação Java ainda não era tão popular como atualmente. Primeiramente o nome do projeto era Xelfi, em alusão ao Delphi, pois, a pretensão deste projeto era ter funcionalidades semelhantes aos IDE's (ambiente de desenvolvimento integrado) então populares do Delphi que eram mais atrativas por serem ferramentas visuais e mais fáceis de usarem, porém com o intuito de ser totalmente desenvolvido em Java. (OFICINA, 2020)

Em 1999 o projeto já havia evoluído para uma IDE proprietário, com o nome de NetBeans DeveloperX2 nome este que veio da ideia de reutilização de componentes que era a base do Java. Nessa época a empresa Sun Microsystems havia desistido de sua IDE Java Workshop e procurando por novas iniciativas adquiriu o projeto NetBeans DeveloperX2 incorporando-o a sua linha de softwares. (OFICINA, 2020)

2.1.2. XAMPP e MySQL

Para fazer a ligação da aplicação com o banco de dados, o uso do XAMPP veio a calhar pois é um software de uso livre perante os termos da GNU. O XAMPP também já inclui o MySQL como banco de dados facilitando o seu uso e pode ser acessado pelas aplicações que fazem seu uso na mesma rede, sendo um pacote com os

principais servidores de código aberto do mercado, incluindo FTP, banco de dados e Apache com suporte às linguagens PHP e Perl.

O MySQL é um SGBD de licença dupla (sendo uma delas de software livre) extremamente popular, de acordo com o ranking mensalmente atualizado da DB-Engines a plataforma só perde em escala mundial para o Oracle em utilização.

Um Sistema de gerenciamento de Banco de Dados ou SGBD é o responsável pelo acesso a uma coleção de dados inter-relacionados, coleção essa conhecida como Banco de Dados, o qual é constituído de dados que são relevantes para uma pessoa, empresa ou entidade. A principal funcionalidade do SGBD é possibilitar o armazenamento e recuperação de tais dados, de forma eficiente, eficaz e conveniente. (SILBERSCHATZ, SUNDARSHAN, & KORTH, 2016)

Os desenvolvedores David Axmark, Allan Larsson e Michael Widenius criaram a solução na década de 90, a fim de satisfazer a necessidade de uma interface compatível com as rotinas utilizadas em suas tabelas e aplicações. Inicialmente foi empregada a API mSQL, porém a velocidade de resposta não bastava, entretanto, a mesma não foi descartada e foi utilizada de base, usando as linguagens C e C++ foi escrita uma nova API que originou o MySQL e posteriormente a empresa responsável por sua manutenção, a MySQLAB. Após essa fase o SGBD transformou-se em um dos mais conhecidos, por possuir particularidades que o tornavam extremamente rápido. Novas versões são lançadas, sempre buscando contemplar novas insuficiências que surgiram ao longo do caminho. (MILANI, 2007)

2.1.3. AndroCode Scanner

Outra tecnologia utilizada foi o APP AndroCode Scanner também gratuito na GooglePlay, que faz a leitura de códigos de barras. É necessário a instalação do App no celular e de um executável no computador que faz a ligação entre os dispositivos. Basta realizar a leitura de um QR Code fornecido pelo executável com o aplicativo do AndroCode Scanner instalado no smartphone, assim o próprio sistema faz a ligação do computador e do celular para poder realizar a leitura do código de barras.

3. CONTEXTO EMPREENDEDOR DA FERRAMENTA

"Você pode pensar que é capaz e fazer alguma coisa e pode pensar que não é. Nos dois casos, você está absolutamente correto"(Henry Ford)

Como foi dito por Maximiano (2012), muitas das pessoas mais prósperas do mundo começaram a vida como pequenos empreendedores, praticamente do zero, como Henry Ford (fundador da Ford e criador da linha de montagem) e Bill Gates, fundador da Microsoft. Pessoas como Ford e Gates transformaram a tecnologia, o modo de fazer negócio e a própria sociedade.

O ato de empreender está diretamente ligado a inovação, a busca de novas oportunidades de negócio, inovar e surpreender em um ambiente que aparentemente todas as boas idéias já foram pensadas e implementadas.

3.1. StartUp

O nascimento de uma empresa de pequeno porte pode ser chamado de Startup, mesmo que a empresa ainda não tenha iniciado suas atividades comerciais.

Startup é uma ideia de empresa ou uma empresa nascente voltado à tecnologia e inovação que tenha como objetivo desenvolver e aprimorar um modelo de negócio. (O que é uma StartUp? | Sebrae, 2020)

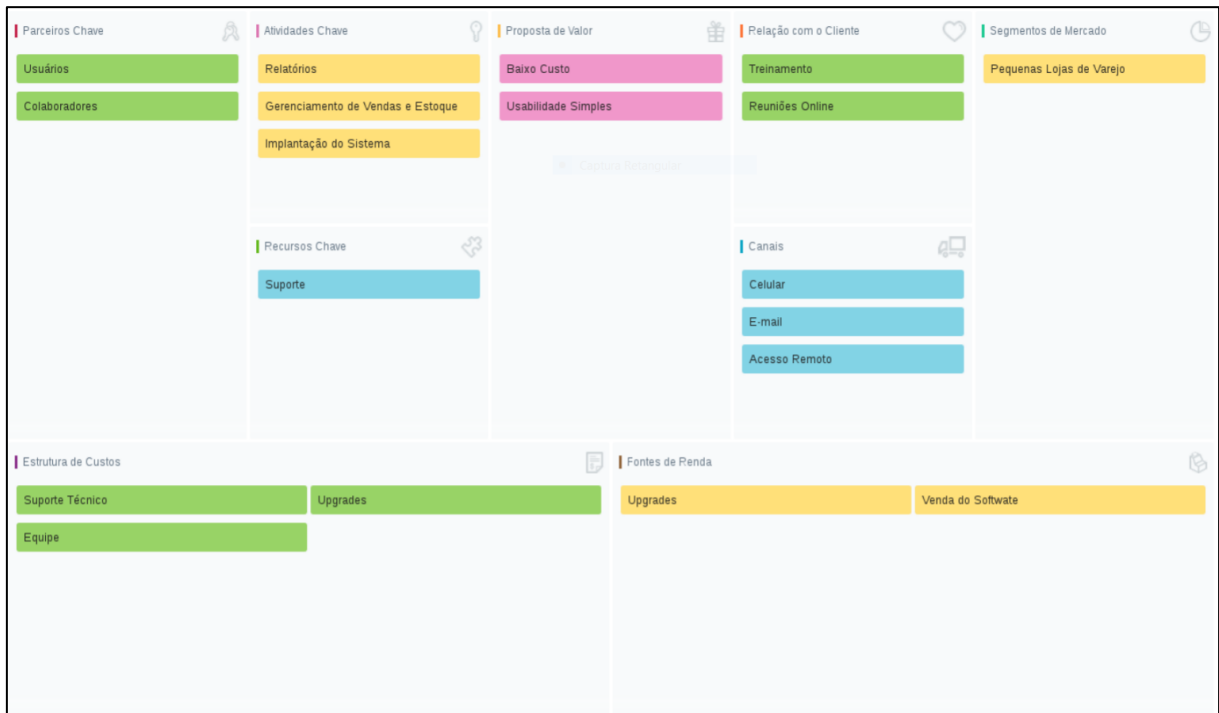
3.2. Canvas

O *Business Model Generation*, conhecido como Canvas, foi desenvolvida pelo Suíço Alex Osterwalder em sua tese de doutorado. O esquema do Canvas permite pessoas co-criarem modelos de negócios levando em consideração nove elementos-chave (proposta de valor, parcerias-chaves, atividades-chaves, recursos-chaves, relacionamento com clientes, segmentos de clientes, canais de distribuição, estrutura de custos e fluxo de receitas). (MOTA, 2020)

MOTA (2020) ainda diz que, o formato da ferramenta Canvas permite analisar visualmente o modelo de negócio em criação, podendo remodelar e adaptar. Esse diferencial possibilita ver o modelo como um desenho e não como uma folha de texto.

A figura 1 mostra de forma detalhada e dinâmica o modelo canvas do projeto.

Figura 1 – Modelo Canvas do projeto



Fonte: O autor.

Descrição dos componentes do modelo Canvas, no contexto do projeto.

1. Segmento de Mercado: O alvo da solução será inicialmente micro e pequenos empreendedores que iniciam seus trabalhos.
2. Proposta de Valor: Propor um sistema de baixo custo acessível ao público alvo e de fácil usabilidade.
3. Canais: O download e instalação da aplicação será feita enviando um link por email ou acesso remoto, onde o cliente entra em contato por email ou telefone.
4. Relação com o Cliente: Será realizado um treinamento online através de video conferencia (que poderá ser utilizado para suporte pós venda) e ainda videos explicativos sobre a plataforma.
5. Fontes de Renda: A venda da aplicação e futuros Upgrades.
6. Recursos Chave: Prestação de Suporte pós venda ao cliente.
7. Atividade Chave: Oferecer ao cliente uma implantação simples e de baixo custo, contendo a emissão de relatórios e controle de vendas/estoque.
8. Parcerias Chave: Os parceiros chave serão futuros colaboradores do projeto, e principalmente os usuarios, que irão fornecer seu feedback.
9. Estrutura de Custos: Suporte fornecido ao cliente, manutenção da equipe e Upgrades da aplicação.

4. ANÁLISE DO PROJETO

A Qualidade é um dos principais fatores observados no desenvolvimento de software, seja ela interna, relacionados na visão do programador, quanto os fatores

de qualidade externa, relacionados a perspectiva dos clientes. Para que haja a garantia da qualidade, faz-se necessário uma gama de processos bem estruturados de desenvolvimento de software. Desta forma, a qualidade de software está intimamente relacionada com a melhoria da qualidade dos processos de desenvolvimento. (ALECRIM, 2020)

4.1. ENGENHARIA DE SOFTWARE

A seguir serão apresentados os processos realizados para desenvolver este projeto juntamente com sua respectiva documentação como diagrama BPMN, diagrama de Caso de Uso, Diagrama de Classe, Requisitos funcionais e não funcionais, 5W1H, TAP e EAP que facilita e auxilia no entendimento de todas as atividades propostas pela solução bem como para realização de alterações/correções que podem surgir.

Os diagramas que não estiverem dispostos neste artigo poderão ser encontrados no repositório do GitHub em Artigo e Documentação/ Documentação do Software (TOMAS, 2020).

4.1.1. BPMN

A Notação de modelagem de processos de negócio (BPMN, em inglês) é um método de fluxograma que modela as etapas, de ponta a ponta, de um processo de negócios planejado. Peça-chave na gestão de processos de negócios, representa de forma visual uma sequência detalhada de atividades de negócios e fluxos de informação necessários para concluir um processo. (LUCIDCHART, 2020)

As figuras 2 e 3 ilustram o BPMN da aplicação, mostrando todo o processo para realizar cadastro e alteração de clientes e realização de uma venda, visto que cadastro e alteração de fornecedores, produtos e usuários serão idênticos ao de cadastro e alteração de clientes.

Figura 2 - Modelo BPMN Alteração e Cadastro de Clientes

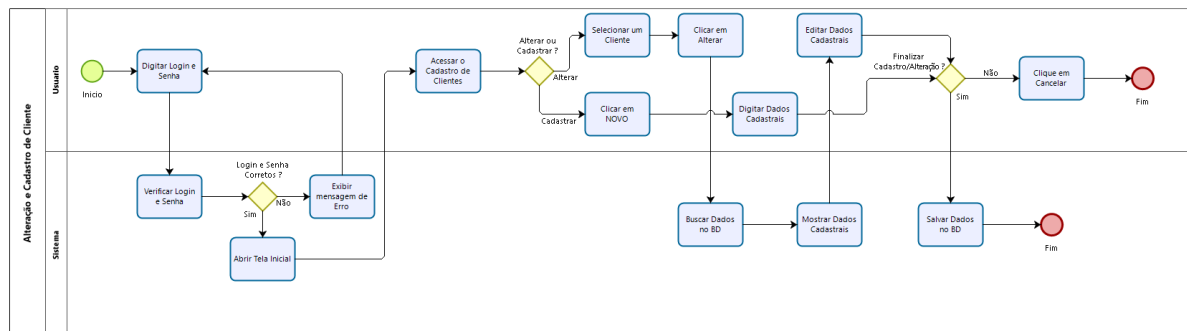
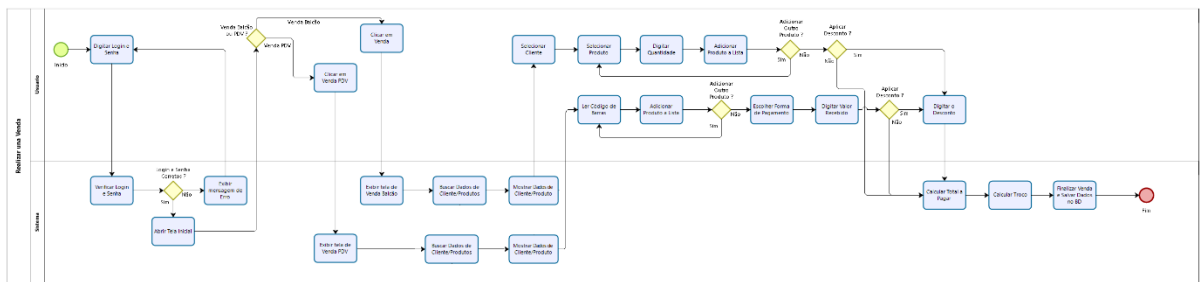


Figura 3 - Modelo BPMN Realizar uma Venda



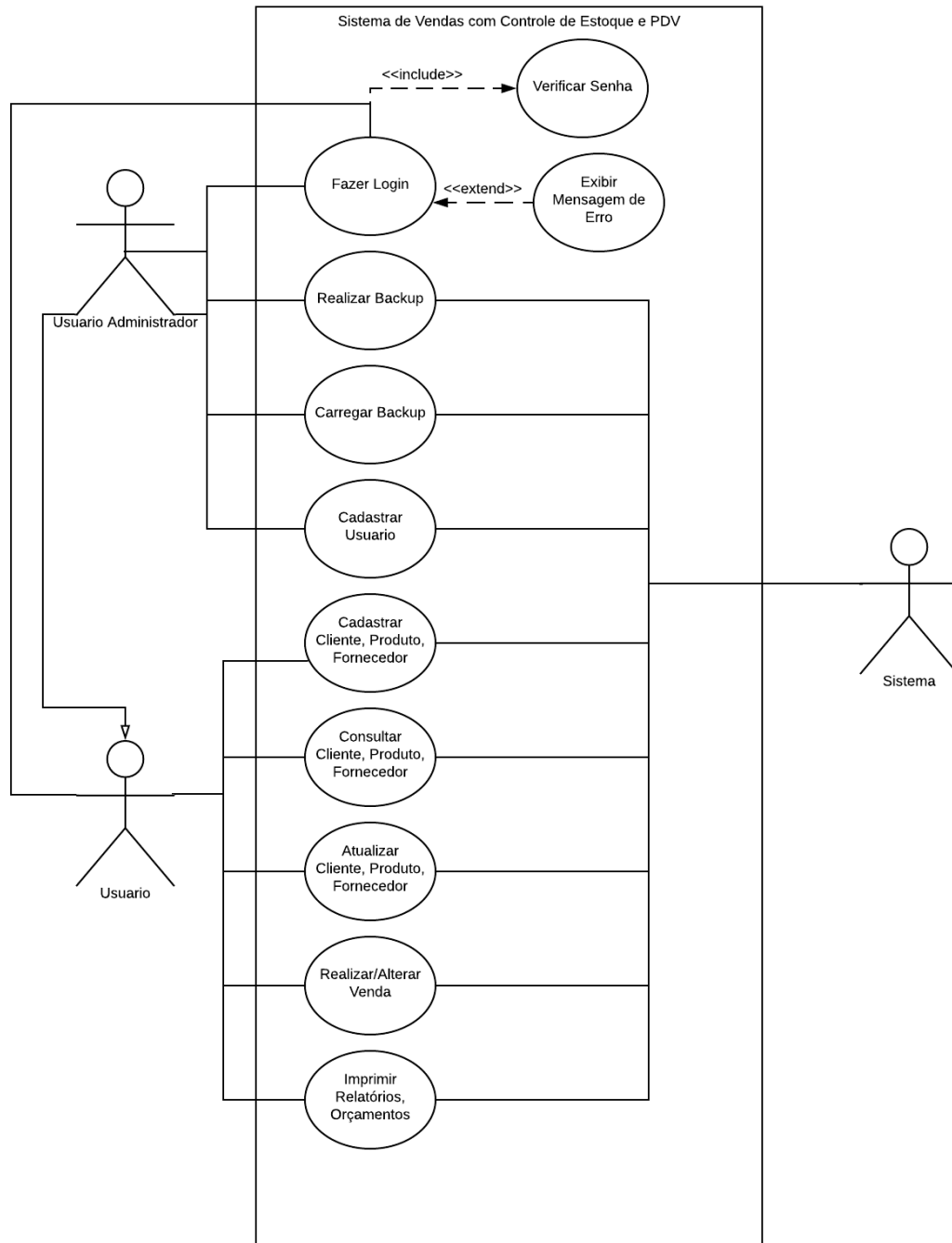
Fonte: O autor.

4.1.2. Diagrama de Caso de Uso

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz. (DEVMEDIA, 2020).

A figura 4 mostra de forma detalhada os papéis desenvolvidos por cada ator na aplicação (Usuário Administrador, Usuário e Sistema)

Figura 4 - Diagrama de casos de uso



Fonte: O autor.

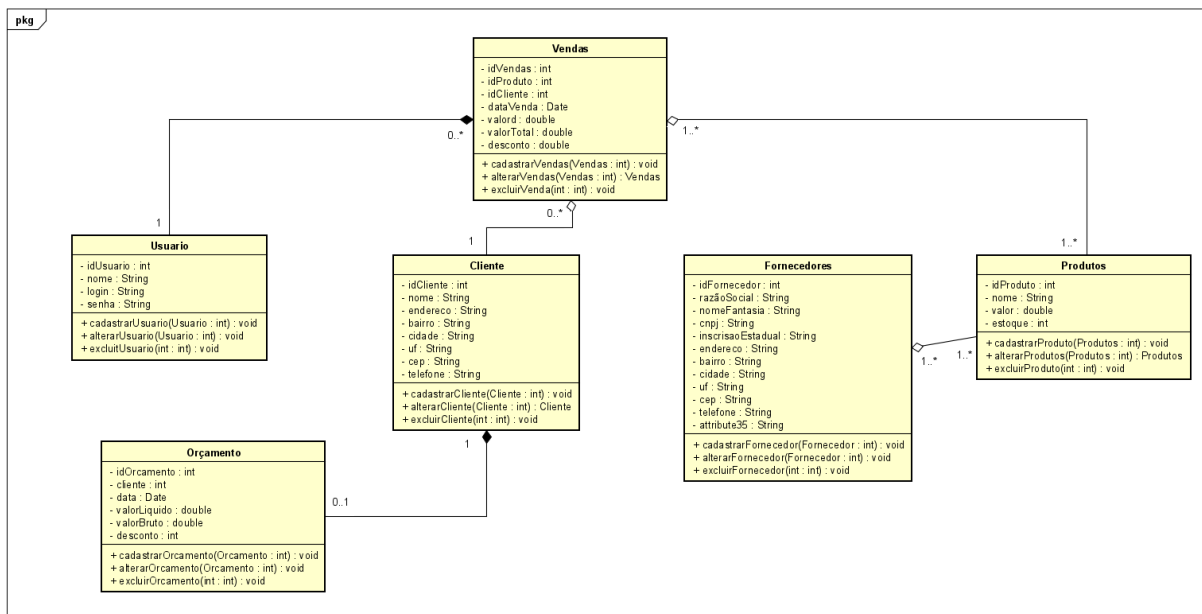
4.1.3. Diagrama de Classe

Em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas

características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes. Na *Unified Modeling Language* (UML) em diagrama de classe, uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos. Abaixo na figura 5 está representado o diagrama de classe do projeto, com todas as suas classes, atributos e métodos. (DEV MEDIA, 2020)

Logo abaixo, a figura 5 mostra o diagrama de classe do projeto.

Figura 5 - Diagrama de Classe



Fonte: O autor.

5. DESENVOLVIMENTO DO APLICATIVO

5.1. TELAS DA APLICAÇÃO E SUAS FUNCIONALIDADES

Abaixo serão apresentadas todas as telas presentes na aplicação desktop Sistema de Venda com Controle de Estoque e PDV juntamente com uma descrição detalhada de cada item presente.

5.1.1. Tela de Login e Autenticação

O primeiro contato que o usuário tem com o sistema é a tela de login e autenticação. Não existe a opção de cadastro de usuário aqui pois o sistema vem com

um usuário administrador como padrão e o primeiro login deve ser feito com o mesmo. Caso o Usuário e Senha estiverem incorretos será exibida uma mensagem de erro, se não, será redirecionado para a próxima tela. A figura 6 é a representação desta tela.

Figura 6 - Tela de Login e Autenticação

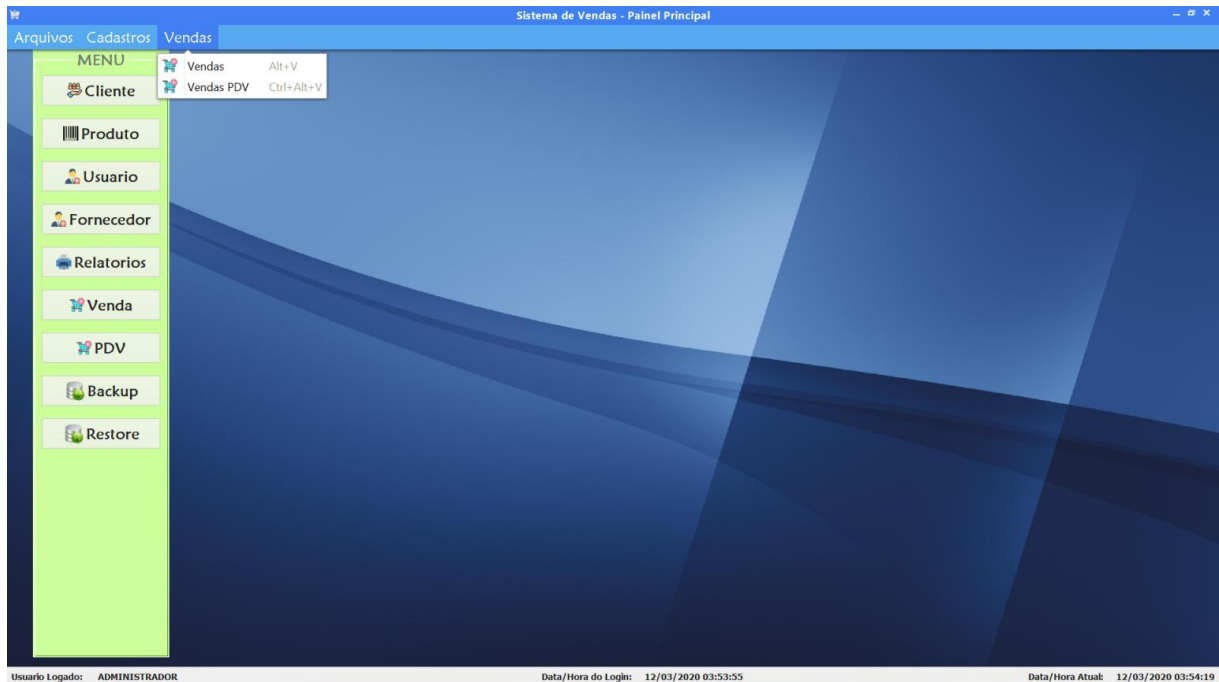


Fonte: O autor.

5.1.2. Tela Principal

Como podemos observar na figura 7, após o login bem-sucedido o usuário é redirecionado para a tela principal do sistema, onde irá encontrar informações como usuário logado, hora do login e hora atual do sistema e todas as principais funcionalidades como Cliente, Produto, Usuário, Fornecedor, Relatórios, Vendas, PDV, Backup e Restore. Vale ressaltar que as Opções de backup e restore só serão visíveis caso o login seja realizado com o usuário administrador.

Figura 7 - Tela Principal



Fonte: O autor.

5.1.3. Tela de Clientes

Na tela de cadastro e busca de clientes, temos uma tabela com todos os clientes cadastrados contendo algumas informações básicas. Também temos a opção de cadastrar um novo cliente clicando no botão “NOVO”, assim habilitando todos os campos a serem preenchidos. Para alterar algum registro, deve-se selecionar um cliente na tabela e clicar em alterar, como pode ser visualizado na figura 8.

5.1.4. Tela de Relatórios

Logo abaixo na figura 9 temos a tela onde podemos fazer a visualização e se necessário a impressão de vários relatórios como, clientes(apenas um cliente e todos os clientes), vendas(todas as vendas, todas as vendas de um cliente e uma venda por cliente) e produtos(todos os produtos ou apenas um produto).

Figura 8 - Tela de Cadastro e Busca de Clientes

Código	Nome	Cidade	UF	Telefone
1	JEFERSON TOMAS DOS SANTOS	FRANCA	SP	(16)99301-0049
2	VALDETE TOMAS DA ROSA SANTOS	SAO JOSE DA BARRA	MG	(35)99167-3021
3	IVALDO MARTINS DOS SANTOS	ALPINÓPOLIS	MG	(35)33167-3021
4	VALERIA TOMAS	FRANCA	SP	(16)33978-4215

Fonte: O autor.

Figura 9 - Tela de Emissão de Relatórios

Codigo da Venda	Data da Venda
1	2020-03-11
3	2020-03-11

Fonte: O autor.

5.1.5. Tela de Cadastro e Busca de Vendas (Realizar uma Venda)

A figura 10 mostra a tela onde temos três abas, a primeira dela é a de cadastro de vendas, que seria para realizar uma “venda balcão” onde o cliente faz o pedido do produto e o atendente faz a checagem se tem o produto em estoque e realiza a venda. Também é possível realizar a emissão de orçamentos, porém esta opção não subtrai o produto do estoque no banco de dados.

Figura 10- Tela de Cadastro e Busca de Vendas (Realizar uma Venda)

Código	Nome	Quantidade	Valor Unitario	Valor Total
--------	------	------------	----------------	-------------

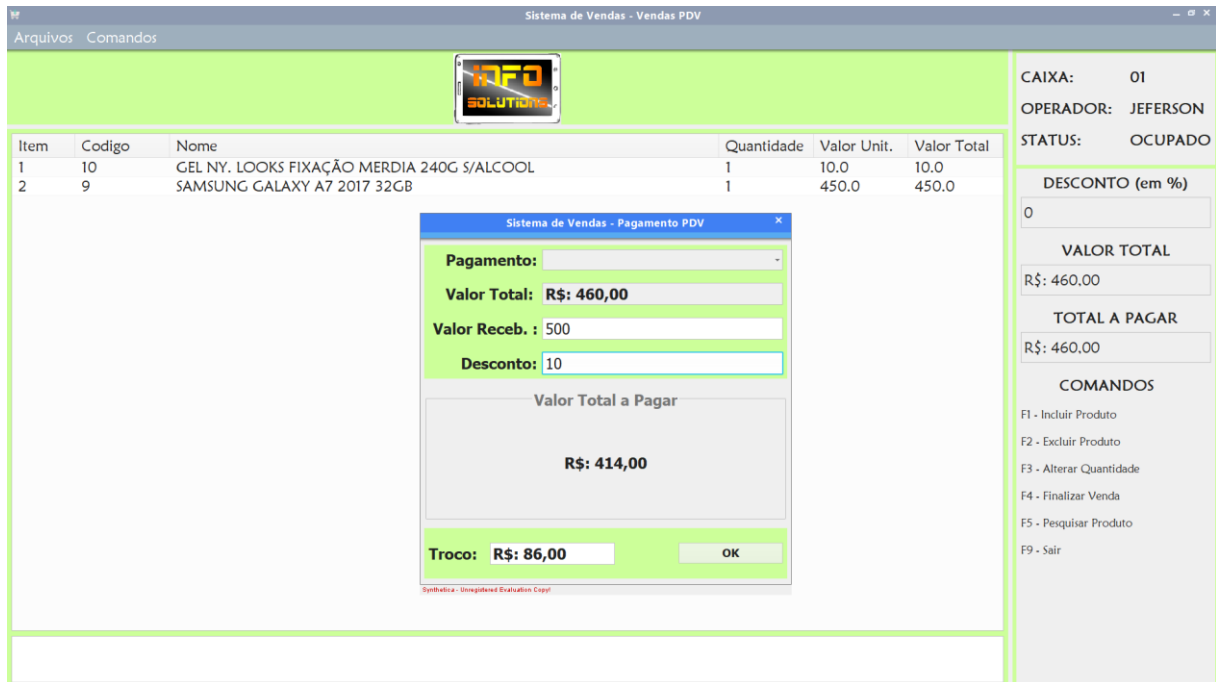
Fonte: O autor.

5.1.6. Tela de Vendas PDV e Pagamento

A tela de vendas PDV, que funciona de uma forma similar ao caixa de um supermercado, onde o cliente leva o produto até o caixa e o(a) atendente faz a leitura do código de barras que irá buscar automaticamente o produto no estoque. Logo após inserir os produtos, o usuário irá para a tela de pagamento onde terá todos os dados mais relevantes como forma de pagamento, valor total, valor recebido, desconto e troco como é apresentado na figura 11.

O sistema possui ao todo onze telas, elas podem ser visualizadas no repositório do GitHub em Artigo e Documentação/ Documentação do Software (TOMAS, 2020)

Figura 11 - Tela de Vendas e Pagamento PDV



Fonte: O autor.

5.2. IMPLEMENTAÇÃO DA APLICAÇÃO

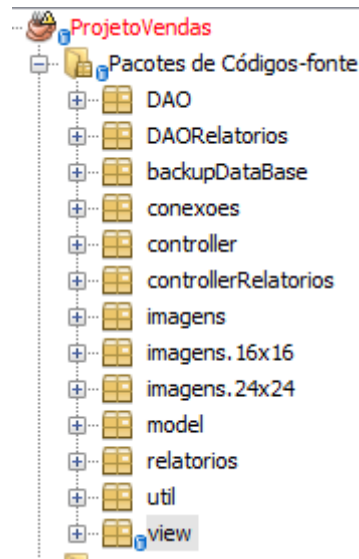
Fazendo uso do modelo de arquitetura MVC o projeto foi dividido em três camadas físicas (Model, View e Controller).

Um componente de visualização renderiza o conteúdo de uma parte particular do Model e encaminha para o Controller as ações do usuário; acessa também os dados do Model via Controller e define como esses dados devem ser apresentados.

Um Controller define o comportamento da aplicação, é ele que interpreta as ações do usuário e as mapeia para chamadas do Model. Em um cliente de aplicações Desktop essas ações do usuário poderiam ser cliques em botões ou seleções de menus. As ações realizadas pelo Model incluem ativar processos de negócio ou alterar o estado do Model. Com base na ação do usuário e no resultado do processamento do Model, o Controller seleciona uma visualização a ser exibida como parte da resposta a solicitação do usuário. Há normalmente um Controller para cada conjunto de funcionalidades relacionadas. (Macoratti, 2020)

Ainda temos o DAO (Objeto de Acesso a Dados) que é um padrão para aplicação que utiliza persistência de dados onde tem a separação das regras de negócio e das regras de acesso ao banco de dados. A figura 12 mostra como é o uso da Arquitetura MVC.

Figura 12 - Arquitetura MVC



Fonte: O autor.

5.2.1. Implementação do Front-End

A interface dos projetos Java possui componentes um pouco diferentes dos presentes no sistema operacional, no que tangem as dimensões, cores, bordas etc. O pacote *javax.swing* possui um conjunto de ferramentas que permite ao programador definir qual será o LookAndFeel da aplicação. Caso não defina nenhum, o próprio *swing* cuidará de selecioná-lo. É apresentado na figura 13 um trecho do código onde ocorre a mudança do LookAndFeel do projeto usando a biblioteca *Synthetic*, na linha 318 mudando o LookAndFeel padrão pelo novo (*SyntheticPlainLookAndFeel*).

O código-fonte referente ao *front-end* do projeto está disponível na íntegra no repositório do GitHub em (TOMAS, 2020)

Figura 13 - Trecho do Código do FrontEnd

```
284 public static void main(String args[]) {
285     /* Set the Nimbus look and feel */
286     Look and feel setting code (optional)
307
308     /* Create and display the form */
309     java.awt.EventQueue.invokeLater(new Runnable() {
310         public void run() {
311
312             //Aplicando LookAndFeel
313             try {
314                 //Configurar se deve aplicar o look a borda do JFrame
315                 // No caso esta programado para não aplicar (Boolean.FALSE)
316                 UIManager.put("Synthetica.window.decoration", Boolean.TRUE);
317                 //Configura qual Look você quer usar no projeto
318                 UIManager.setLookAndFeel(new SyntheticaPlainLookAndFeel());
319             } catch (ParseException | UnsupportedLookAndFeelException erro) {
320             }
321             //Termina o LookAndFeel
322
323             new Login().setVisible(true);
324         }
325     });
```

Fonte: O autor.

5.2.2. Implementação do Back-End

Hoje, as aplicações são desenvolvidas através de interfaces amigáveis para que os usuários consigam trabalhar e manipular informações no banco de dados de forma facilitada. Geralmente, muitos programadores estudam o mecanismo do *JDBC* - *Java Database Connectivity* que ajudam nas aplicações com bibliotecas de baixo nível e até mesmo servem de base para as de alto nível. Pode-se dizer que é uma API que reúne conjuntos de classes e interfaces escritas na linguagem Java na qual possibilita se conectar através de um driver específico do banco de dados desejado. Com esse driver pode-se executar instruções SQL de qualquer tipo de banco de dados relacional. Para fazer a comunicação entre a aplicação e o SGBD é necessário possuir um driver para a conexão desejada. Geralmente, as empresas de SGBDs oferecem o driver de conexão que seguem a especificação JDBC para caso de algum desenvolvedor querer utilizar. (Thiago, 2020)

Na Figura 14, é apresentado um trecho do código BackEnd da aplicação que faz a conexão com o banco de dados, das linhas 37 à 42 a função recebe como parametro o servidor, nome do banco, usuário e senha. Nas linhas 54 e 55 é onde ocorre a conexão com o banco de dados. O código-fonte referente ao *front-end* do projeto está disponível na íntegra em (TOMAS, 2020).

Figura 14 - Trecho do Código do BackEnd

```
37 public ConexaoMySQL(String pServidor, String pNomeDoBanco, String pUsuario, String pSenha){
38     this.servidor = pServidor;
39     this.nomeDoBanco = pNomeDoBanco;
40     this.usuario = pUsuario;
41     this.senha = pSenha;
42 }
43
44 /**
45  * Abre uma conexao com o banco
46  * @return Connection
47  */
48 public Connection conectar(){
49     try {
50         //Driver do PostgreSQL
51         Class.forName("com.mysql.jdbc.Driver").newInstance();
52
53         //local do banco, nome do banco, usuario e senha
54         String url = "jdbc:mysql://" + servidor + "/" + nomeDoBanco;
55         this.setCon((Connection) DriverManager.getConnection(url, usuario, senha));
56
57         //se ocorrer tudo bem, ou seja, se conectar a linha a seguir é executada
58         this.status = true;
59     }
60     catch(ClassNotFoundException | InstantiationException | IllegalAccessException | SQLException e){
61         JOptionPane.showMessageDialog(null, e.getMessage());
62     }
63     return this.getCon();
64 }
```

Fonte: O autor.

6. CONCLUSÃO

Concluída a implementação do *Minimum Viable Product* (MVP) da aplicação desktop sistema de vendas com controle de estoque, que possibilita que pequenos empreendedores possam gerenciar seu negócio de forma simples e dinâmica, o objetivo do projeto foi atingido com sucesso.

O trabalho foi realizado a fim de colocar em prática todo o conhecimento adquirido ao longo do curso, utilizando o SDK (*Software Development Kit*) do Java e a IDE (*Integrated Development Environment*) NetBeans, conceitos e artefatos da Engenharia de Software, e por fim, ferramentas de métodos ágeis para acompanhamento do desenvolvimento do projeto, como o Canvas.

O valor para comercialização do software manterá uma média de R\$20,00 mensais, podendo mudar de acordo com a necessidade e contrato firmados com o cliente, levando em consideração que futuras implementações serão feitas com intuito de trazer mais funcionalidades que podem beneficiar o usuário. Podemos destacar a impressão de cupom fiscal e a visualização de gráficos diversos (vendas realizadas, produtos mais vendidos etc.)

Com a real aplicação do sistema, será possível verificar se as funcionalidades estão usadas de forma efetiva assim podendo levantar requisitos para futuras melhorias. A intenção é implementar o sistema em um negócio de pequeno porte durante o período de 60 dias, assim recebendo um *feedback* do empreendedor,

podendo fazer uma análise e verificar se é viável expandir o projeto tornando-se um negócio lucrativo.

Referências

- ALECRIM, E. (09 de 06 de 2020). *Processos de engenharia de software*. Fonte: administradores.com: <https://administradores.com.br/artigos/processos-de-engenharia-de-software>
- DEITEL, P., & DEITEL, H. (2010). *Java Como Programar* (8ª ed.). Editora Pearson. Acesso em 15 de Março de 2020, disponível em <https://plataforma.bvirtual.com.br/Leitor/Publicacao/1142/pdf/37>
- DEITEL, P., & DEITEL, H. (2016). *Java Como Programar* (10ª ed.). Editora Pearson. Acesso em 15 de Março de 2020, disponível em <https://plataforma.bvirtual.com.br/Leitor/Publicacao/39590/pdf/47>
- DEVMEDIA. (09 de 06 de 2020). *O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML*. Fonte: DevMedia: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>
- DEVMEDIA. (10 de 06 de 2020). *Orientações básicas na elaboração de um diagrama de classes*. Fonte: DevMedia: <https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>
- LUCIDCHART. (09 de 06 de 2020). *O que é BPMN*. Fonte: Lucidchart: <https://www.lucidchart.com/pages/pt/o-que-e-bpmn>
- Maximiano, A. C. (2012). *Empreendedorismo* (1ª ed.). São Paulo: Editora Pearson. Acesso em 16 de Março de 2020, disponível em <https://plataforma.bvirtual.com.br/Acervo/Publicacao/3019>
- MILANI, A. (2007). *MySQL Guia do Programador*. Editora Novatec.
- MOTA, G. (16 de Março de 2020). *Canvas: O que é e para que serve ?* Fonte: Administradores: <https://administradores.com.br/artigos/canvas-o-que-e-e-para-que-serve>
- O Que é um ERP ? Oracle Brasil*. (15 de Março de 2020). Fonte: Oracle Brasil: <https://www.oracle.com/br/applications/erp/what-is-erp.html#link7>
- O que é uma StartUp? | Sebrae*. (16 de Março de 2020). Fonte: Sebrae: <https://www.sebrae.com.br/sites/PortalSebrae/ufs/ac/artigos/o-que-e-uma-startup,17213517aa47a610VgnVCM1000004c00210aRCRD>

SILBERSCHATZ, A., SUNDARSHAN, S., & KORTH, H. F. (2016). *Sistema de banco de dados*.

TOMAS, J. (09 de 06 de 2020). *JefersonTS/TCC-Projeto-PDV*. Fonte: github.com:
<https://github.com/JefersonTS/TCC-Projeto-PDV>